2013 IEEE International Conference on Robotics and Automation (ICRA)
Karlsruhe, Germany, May 6-10, 2013

Autonomous Robotic Valve Turning: A Hierarchical Learning Approach

Seyed Reza Ahmadzadeh, Petar Kormushev and Darwin G. Caldwell

***Abstract*— Autonomous valve turning is an extremely challenging task for an Autonomous Underwater Vehicle (AUV). To resolve this challenge, this paper proposes a set of different computational techniques integrated in a three-layer hierarchical scheme. Each layer realizes specific subtasks to improve the persistent autonomy of the system. In the first layer, the robot acquires the motor skills of approaching and grasping the valve by kinesthetic teaching. A Reactive Fuzzy Decision Maker (RFDM) is devised in the second layer which reacts to the relative movement between the valve and the AUV, and alters the robot's movement accordingly. Apprenticeship learning method, implemented in the third layer, performs tuning of the RFDM based on expert knowledge. Although the long-term goal is to perform the valve turning task on a real AUV, as a first step the proposed approach is tested in a laboratory environment.**

I. INTRODUCTION

With the advent of robust guidance systems, advanced processing capabilities and high capacity lightweight batteries, nowadays, AUV robots are becoming the platform of choice for conducting underwater missions with ever increasing duration and complexity. Implementing innovative machine learning methods enhances capabilities such as operating under extreme uncertainty, interacting with highly unexpected underwater environment autonomously, learning from sensor data continuously, and re-planning the mission optimally.

The European project 'Persistent Autonomy through learNing, aDaptation, Observation and Re-plAnning' (PANDORA) [1] aims to make underwater robot persistently autonomous by developing and evaluating new computational methods. Autonomous grasping and turning a valve is one of the most challenging tasks which has been defined in PANDORA, and has been under investigation since the start of the project in January 2012. Coincidentally, the recently announced DARPA robotics challenge also includes as one subtask an autonomous valve turning scenario, although it is not in underwater environment. Despite the numerous challenges of the underwater environment, we believe that autonomous valve turning is within the reach of existing state of the art in robotics.

In PANDORA, the valve turning task will be accomplished by Girona500 [2], which is a compact and lightweight AUV with hovering capabilities, reconfigurable propulsion system, and mission-specific payloads (a robotic arm in this case). Before attempting the valve turning task underwater, an integrated system has been built in the lab including an Optitrack system and a lightweight KUKA-DLR robotic arm. The Optitrack system captures real-time 3D position and orientation data of a rigid body using a number of motion capture cameras and a set of markers. The Optitrack system simulates the AUV's sensors in our experiment. An important feature of this system is that, it provides precise and high frequency data, whereas the real AUV is equipped with different type of sensors, e. g., stereo camera and gyro-enhanced Attitude and Heading Reference System (AHRS). The KUKA-DLR robotic arm is used under Cartesian impedance control mode [3]. The robot's position, orientation and joint or Cartesian stiffness commands are sent to KUKA controller using the DLR's Fast Research Interface libraries [4]. Although the KUKA-DLR is a different kind of manipulator than the one is attached to the Girona500, the proposed trajectory generation learning method is not dependent on the kinematics of the manipulator. Underwater robotic valve turning consists of two main steps. Firstly, the robot approaches the valve while the internal control system stabilizes the system counteracting reaction forces from underwater turbulence and other disturbances. Secondly, the arm is actuated to grasp and turn the valve. By assuming that the first step is accomplished, the goal of this paper is to develop the second step. Another assumption considered in this paper is that, in underwater environment, the valve is fixed to the panel and the AUV moves, while in the lab we move the valve and keep the robot base fixed. In our experiment, the turning phase which starts after reaching and grasping phases, is hardcoded to the robot. Our future work includes analysing the force/torque sensor data on the end-effector and developing a learning system to turn the valve.

II. RELATED WORK

Robotic valve manipulation was found to contain a number of complex and challenging subtasks. Consequently, there seem to be few published description of attempts directly related to underwater tasks. Prior works in industrial robotic valve operation, generally use nonadaptive classical control and basic trajectory planning methods. Using a six degree-of-freedom industrial robot equipped with a number of sensors (e. g., vision, range, sound, proximity, force/torque, and touch) Abidi et al. try to achieve inspection and manipulation capabilities in the semi-autonomous operation of a control panel in a nuclear power plant [5]. The main drawback is that this approach is developed for static environments with

This research was sponsored by the PANDORA [1] EU FP7 project under the grant agreement No. ICT-288273.

S. R. Ahmadzadeh, P. Kormushev, and D. G. Caldwell are with the Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy. {reza.ahmadzadeh, petar.kormushev, darwin.caldwell} at iit.it

978-1-4673-5642-8/13/$31.00 ©2013 IEEE

predefined dimensions and scales. For instance, the size and position of the panel, the valve, and other objects in the room are manually engineered into the system. More recent approaches generally use sensor-based movement methods which implies that the robot trajectories have not been programmed off-line [6]. The robot, which is equipped with a torque sensor, detects the valve handle which is equipped with a proximity sensor. The authors focus on a model-based approach to sense and to avoid over-tightening/loosening of the valve. The other phases are accomplished using classical methods. In underwater domain, still Remotely Operated underwater Vehicles (ROV) are being used due to the complexity and uncertainty of the environment. These ROVs are operated by one or two skilled operators, usually one keeps the robot stable while the other controls the manipulator [7]. Machine learning methods have received special attention in recent years. The key idea is that, although the above mentioned approaches cannot deal with unknown dynamic environments (e.g. underwater environment), using new learning methods the robot can acquire new skills to overcome the uncertainties presented by the real world. Some skills can be successfully transferred to the robot using imitation strategies [8], [9] the others can be learned very efficiently by the robot using reinforcement learning [10]. This realization leads further developments in autonomous AUV valve turning which is a focus of our previous work [11]. we presented an approach to transfer the trajectory generating motor skill to a robotic arm using imitation learning. In addition we developed a fuzzy controller to solve the collision avoidance problem.

In this paper we propose a hierarchical approach to achieve autonomous valve turning using machine learning methods.

III. HIERARCHICAL ARCHITECTURE

The proposed approach is organized as a hierarchical architecture with three different layers which are illustrated as a high level outline in Fig. 1. Each layer realizes specific subtasks to improve the persistent autonomy of the system. The lowest layer is responsible for evaluating demonstrations and generating smooth trajectories using learning methods. In this layer an integrated approach is used which allows the robot-arm to obtain new motor skills by kinesthetic teaching [12]. Imitation learning [8] is used for training the manipulator to learn positional profile. The middle layer is responsible for evaluating relative movements and supervising the subordinate layer. Observing the feedbacks from the Optitrack sensor, this upper layer provides prior decisions depending on the relative behavior of the valve which affects the dynamics of the system. A reactive fuzzy system, RFDM, is established for producing proper decisions based on linguistic rules. The RFDM reacts to the relative movement between the AUV and the valve dynamically and alters the generated trajectory in the lower layer accordingly. The highest layer, is responsible for tuning the parameters of the RFDM system using the expert knowledge. Four various local and global optimization algorithms are implemented to find the best optimum solution.

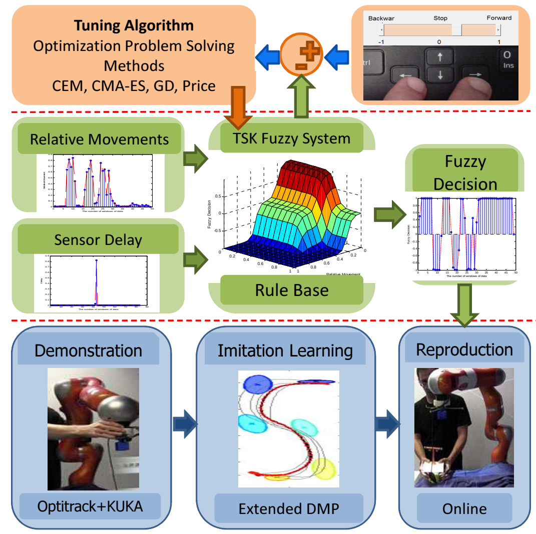

Fig. 1. A high-level diagram illustrating the three layers of the proposed hierarchical learning approach.

IV. KINESTHETIC TEACHING

This approach consists of three consecutive phases: demonstration, imitation learning, and reproduction.

A. Demonstration Phase

The proposed methodology for demonstrating the skill is based on: (i) the joint-space of the KUKA-DLR manipulator in gravity-compensated mode; (ii) real-time 3D position and orientation data derived by Optitrack system; (iii) a mock-up (T-bar shaped) valve to simulate the real valve. A reference and a relative coordinate systems are placed on the valve and the end-effector of the robot, respectively. According to this setting, one demonstration is defined as: moving the end-effector from an arbitrary initial position towards the valve, so that the relative and reference coordinate systems coincide. The position, velocity, and acceleration data of the end-effector are continuously recorded in the robot's frame of reference using the Optitrack system. The recorded demonstrations are shown in Fig. 2.

B. Imitation Learning Phase

During this phase we apply an extension of Dynamic Movement Primitives (DMP) [9] to learn a compact representation of the reaching skill using the recorded demonstrations. The applied approach [13] encapsulates variation and correlation information across multi-variable data. A set of virtual attractors is used in this method to reach a target. The influence of these virtual attractors is smoothly switched along the movement on a time basis. The set of attractors is learned by weighted least-squares regression, by using the residual errors as covariance information to estimate stiffness gain matrices. A proportional-derivative controller is used to move sequentially towards the sequence of targets.

The positional constraints of the demonstrated skill are represented as a mixture of dynamical systems that encode robustly the position trajectory. In this method a full matrix K_i^P associated with each of the K primitives is considered instead of a fixed κ^P gain. The variability and correlation information along a movement can be taken into consideration for learning and reproduction. In our experiment, M examples of the skill are demonstrated in slightly different situations. Each demonstration $m \in \{1...M\}$ consists of a set of T_m positions x, velocities $\dot{x}$ and accelerations $\ddot{x}$ of the end-effector in Cartesian space, where $x \in \Re^3$. A dataset is formed by concatenating the $N = \sum_{m=1}^{M} T_m$ datapoints $\left\{ \{x_j, \dot{x}_j, \ddot{x}_j\}_{j=1}^{T_m} \right\}_{m=1}^{M}$. A desired acceleration is computed based on a mixture of K proportional-derivative systems.

$$\hat{\ddot{x}} = \sum_{i=1}^{K} h_i(t) \left[K_i^P (\mu_i^x - x) - \kappa^\nu \dot{x} \right] \tag{1}$$

In this method, the superposition of basis vector fields is determined by an implicit time dependency. And a decay term defined by a canonical system $\dot{s} = -\alpha.s$ is used to create the implicit time dependency $t = -\ln(s)/\alpha$, where s is initialized with $s = 1$ and converges to zero. Also for the backward movement, which is used in retracting mode, a complementary equation is used to generate time starting from final time to initial time. Furthermore, a set of Gaussians $N(\mu_i^T, \Sigma_i^T)$ is defined in time space τ, with centers μ_i^T equally distributed in time, and variance parameters Σ_i^T set to constant value inversely proportional to the number of states. α is initially fixed depending on the duration of the demonstrations. By determining the weights h_i through the decay term s, the system sequentially converges to the set of attractors in Cartesian space defined by centers μ_i^T, and stiffness matrices K_p are learned from the observed data, either incrementally or in a batch mode.

C. Reproduction Phase

Finally, a stand-alone reproduction of the task can be performed using the learned positional profiles of the task from any arbitrary initial position of the end-effector (Fig. 2). Parts of the movement where the variations across the different demonstrations are large, indicate that the reference trajectory does not need to be tracked precisely. On the other hand, parts of the movement exhibiting strong invariance across the demonstrations should be tracked precisely, i.e., the stiffness used to track the position errors needs to be high.

V. REACTIVE FUZZY DECISION MAKER

To accomplish the approaching phase and grasp the valve, the AUV needs to cope with the undesirable and inevitable relative movements between the robot and the valve which is caused by external disturbances such as underwater currents. Either the sensor causes a delay or the relative movement exceeds a normal range, the robot may miss the valve or break it off. Based on our previous work in [11], a RFDM system is developed in the second layer to observe the dynamic condition and generate a decision command accordingly. The developed RFDM system takes two inputs:

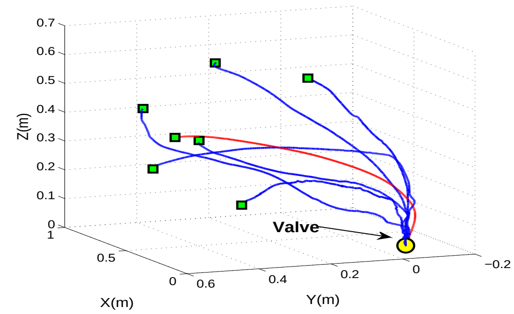

Fig. 2. Six demonstrations (in blue) starting from different initial positions (the green squares), and one reproduction (in red) from a new initial position.

the estimated relative movement between the valve and the end-effector, and the time delay since last sensor update. The RFDM estimates the dynamical behavior of the relative movement over a frame of received data. In addition, it evaluates certainty of the current situation due to the time delay of the sensor. If the delay is *big* then the uncertainty of the current situation of the valve is *high* and vice versa. Since the duration of the movement in the reproduction phase is not fixed and the trajectory is time-independent, the movement itself changes when the RFDM reacts to the relative movement of the valve.

The output of the RFDM system is a continuous numeric command in the range [-1,1], where -1, 0, and 1 correspond to maximum speed retracting, waiting, and maximum speed approaching the valve, respectively. The TSK fuzzy system [14], including product inference engine, singleton fuzzifier, and center average defuzzifier, is used to develop the RFDM. Three Gaussian fuzzy sets for each input variable and three constant outputs are defined. The constructed rule base of the system comprises nine rules. The rule base is complete, continuous, and consistent, and is reported in Table I. The advantage of using fuzzy systems is that they are based on linguistic rules and the parameters that specify membership functions have clear physical meanings and there are methods to choose good initial values for them [14].

TABLE I
FUZZY RULE BASE

		Relative Movement		
		Small	Medium	Big
Sensor Delay	Low	Forward	Stop	Backward
	Medium	Forward	Stop	Backward
	High	Stop	Backward	Backward

If y_i is the output level of each rule weighted by the firing strength w_i of the rule, the final output of the system is the weighted average of all rule outputs, computed as:

$$f_{RFDM} = \sum_{i=1}^{N} w_i y_i \Big/ \sum_{i=1}^{N} w_i \tag{2}$$

VI. TUNING BY APPRENTICESHIP LEARNING

Generally, the human knowledge about a particular engineering problem can be classified in two categories: conscious knowledge and subconscious knowledge. The conscious knowledge is the knowledge that can be explicitly expressed in words or numbers. (e.g. in section V, the conscious knowledge of the expert tutor is used to design the fuzzy rule base.) By the subconscious knowledge we refer to the situation where the human expert knows what to do but cannot express exactly in words how to do it. In our approach, we use apprenticeship learning to derive the subconscious knowledge of the expert for tuning the developed RFDM system. This tuning process is placed in the highest layer of the proposed hierarchical structure.

The designed RFDM in section V contains 6 Gaussian membership functions (Z), 9 linguistic rules, and 3 constant outputs (y). Each Gaussian membership function, $Z = exp(-((x-c)/\sigma)^2)$, includes two tunable parameters: a center (c) and a variance (σ). Since we defined the linguistic rules according to the desired physical behavior of the fuzzy system (conscious knowledge), the other 15 parameters (6 centers, 6 variances, and 3 constant outputs) are considered to be tuned.

After the kinesthetic teaching phase is accomplished and the first layer is capable of generating new trajectories, a tutor simulates the effect of the underwater currents by oscillating the valve in different Cartesian directions in specific times. Simultaneously, using a slider button, another expert tutor can supervise the robot arm while it is following the reproduced trajectory. The tutor applies appropriate continuous commands to the system in the range [-1 1] (-1 means go backward along the trajectory with 100% speed and 1 means go forward along the trajectory with 100% speed). For instance, when the valve is oscillating with a *big* amplitude, the tutor smoothly moves the slider backwards to retract the arm and prevent it from any collision. All data, including the position of the end-effector and the valve, and the tutor's commands are recorded during the apprenticeship learning process. The recorded data is used to tune the RFDM in off-line mode. The tuning task can be done using optimization algorithms.

In the rest of this section, we implement a number of local and global optimization algorithms to this tuning task. To be consistent, in all of the implementations, the error between the recorded data from the tutor and the output of the the untuned fuzzy system is used to make the objective function.

A. Gradient Descent (GD) Method

The GD method is a first-order local optimization algorithm. GD is based on the observation that if a multi-variable function, $J(\theta)$, is defined and differentiable in a neighborhood of a point θ_0, then the function decreases fastest if one goes from θ_0 in the direction of the negative gradient of $J(\theta)$ at θ_0. And we have:

$$\theta_{n+1} = \theta_n - \alpha \nabla J(\theta_n) \tag{3}$$

where in (3) $\alpha \in [0,1]$ is the learning rate. We consider the objective function to be:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{N}(f_{RFDM}(\theta) - f_{SKE}) \tag{4}$$

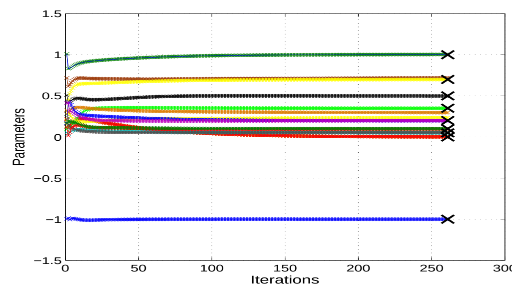

Fig. 3. Convergence of the RFDM parameters using GD method.

where in (4) f_{RFDM} and f_{SKE} are the output of the untuned RFDM system and the recorded subconscious knowledge of the tutor respectively. θ is the set of the parameters to be tuned including the centers and the variances of the fuzzy membership functions, and the constant outputs ($\theta = \{c, \sigma, y\}$). The gradient of the defined objective function with respect to the parameters in θ are calculated in the form of following equation:

$$\frac{\partial J}{\partial \theta_i} = \frac{\partial J}{\partial f_{RFDM}} . \frac{\partial f_{RFDM}}{\partial Z_i} . \frac{\partial Z_i}{\partial \theta_i}, \quad \theta \in \{y_i, c_i, \sigma_i\} \tag{5}$$

where, $i = 1 \ldots 15$. Applying GD method to the set of equations in (5), the algorithm minimizes the objective function. The behavior of the GD algorithm drastically depends on the initial guess. As shown in Fig. 3, in this case the algorithm converges to the optimal solution very fast (200-300 iterations) by choosing a proper initial guess. However, the algorithm may converge very slowly (10000-15000 iterations) or even diverge using other initial conditions.

B. Cross Entropy (CE) Method

The CE method is a generic approach to combinatorial and multi-external optimization and rare event simulation [15]. The CE method involves an iterative procedure where each iteration can be broken down into two phases: (i) Generate a random data sample according to a specified random mechanism, (ii) Update the parameters of the random mechanism based on the data to produce a 'better' sample in the next iteration. The significance of the CE method is that it defines a precise mathematical framework for deriving fast and in some sense 'optimal' updating/learning rules, based on advanced simulation theory [16].

We use a variation of initial sample size between 50 and 400 for the first sample generation, whereas the next generation samples are provided using the best 10% of the previous samples. Depending on the initial mean and variance values of the samples, the algorithm converges to a local optimal solution in 50 - 500 iterations (see Fig. 4).

C. Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

The CMA-ES is a stochastic, derivative-free method for real-parameter (continuous domain) optimization of nonlinear, nonconvex optimization problems [17]. In each generation (iteration) new individuals (candidate solutions) are

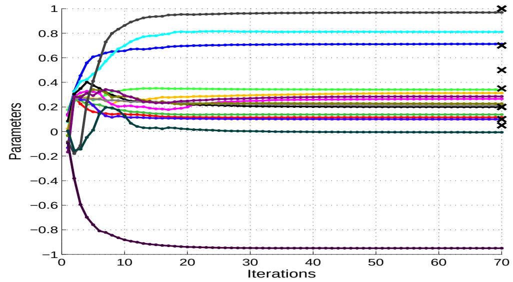

Fig. 4. Convergence of the RFDM parameters using CE method.

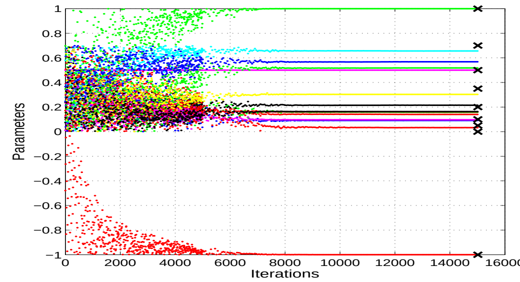

Fig. 6. Convergence of the RFDM parameters using Modified Price's method.

generated by sampling a multi-variable normal distribution. In the next step some individuals are selected for the next generation based on their fitness or objective function value. Pairwise dependencies between the variables in this distribution are represented by a covariance matrix. The CMA is a method to update the covariance matrix of this distribution. Over the generation sequence, individuals with better fitness are generated. This optimization algorithm is particularly useful if the objective function is ill-conditioned. In contrast to most other evolutionary algorithms, the CMA-ES is quasi parameter-free. However, the number of candidate samples (population size) can be adjusted by the user in order to change the characteristic search behavior [17].
The algorithm originally uses the population size equal to 8. For our optimization task with 15 parameters to tune, the algorithm converges around 10000 iterations (see Fig. 5). However, increasing the population size (e.g. by 3 times) decreases the number of iterations (to 3000-4000). Amongst the implemented algorithms, the CMA-ES is the fastest in terms of computation time, and requires a small number of initial parameter settings.

D. Modified Price's Algorithm

This algorithm is designed for particularly difficult global optimization problems in which the evaluation of the object function is very expensive, and the derivatives of the objective function are not available. To solve this problem, we implemented a modified Price's algorithm by Brachetti et al. [18]. The modified Price's algorithm is a population-based algorithm with global and local search parts. The global search part consists of the weighted centroid and the weighted reflection. The Number-theoretic method is applied to generate the initial population and a simplified quadratic approximation using the three best points is adopted, instead of the quadratic model of the objective function in original Price's algorithm. There is just one initial parameter for this algorithm to be set and that is the size of population.
As depicted in Fig. 6, the algorithm converges around 15000 iterations with a population size of 300. The algorithm converges very slowly even if the population size is increased to 1500.

E. Results and Comparison

Assigning obtained values from each optimization algorithm to our designed RFDM structure, we acquire 4 slightly different RFDM structures. As an illustration, the output surface of each attained system is shown in Fig. 7 (top). In addition, the tuned Gaussian membership functions by each algorithm are given in Fig. 7 (bottom). Finally, Root-Mean-Square Error (RMSE) between each two solutions is calculated and reported in Table II. One can see that, the closest solutions to the reference subconscious knowledge of the tutor are obtained from the CMA-ES and the GD methods.

TABLE II
RMSE TABLE

	GD	CEM	CMA-ES	Price
Ref	0.0263	0.1015	0.0044	0.1665
GD	—	0.1067	0.0285	0.1558
CEM	—	—	0.0994	0.1955
CMA-ES	—	—	—	0.1699

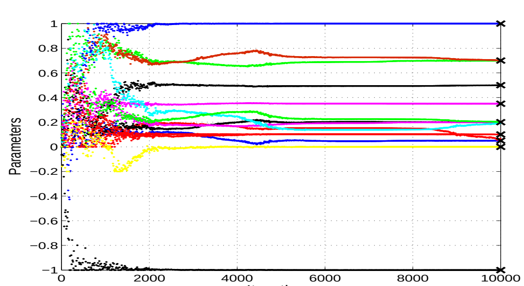

Fig. 5. Convergence of the RFDM parameters using CMA-ES method.

VII. COMPLETE EXPERIMENT

The complete autonomous robotic valve turning experiment using the proposed hierarchical learning approach is summarized as following steps:

- Recording a set of demonstrations from various arbitrary initial positions (According to IV-A).
- Applying proposed Extended DMP method to learn the new motor skill (According to IV-B).
- Recording the expert knowledge, while robot is in reproduction mode (According to IV-C and VI).

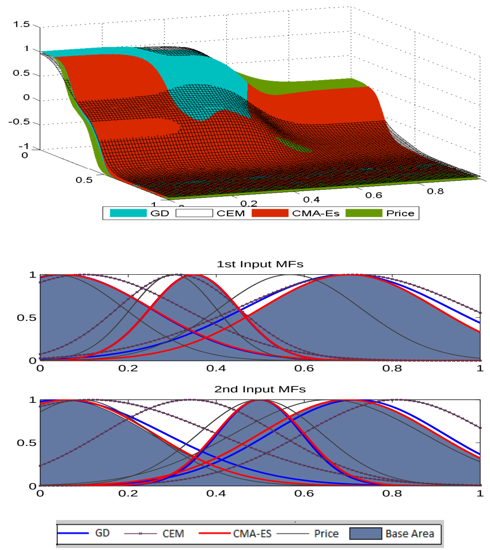

Fig. 7. Final fuzzy surfaces (top) and tuned membership functions (bottom)

- Tuning the RFDM system using the recorded data from previous step (According to V and VI).
- Accomplishing reproduction mode and grasping the valve using tuned RFDM.
- Applying a torque control on end-effector an turn the valve (This part is hardcoded to the robot).

The set of images in Fig. 8 (top) represents five steps of the experiment. The recorded position of the end-effector in a complete experiment is shown in Fig. 8 (bottom). A video accompanying this paper which shows different phases separately is available online at [19].

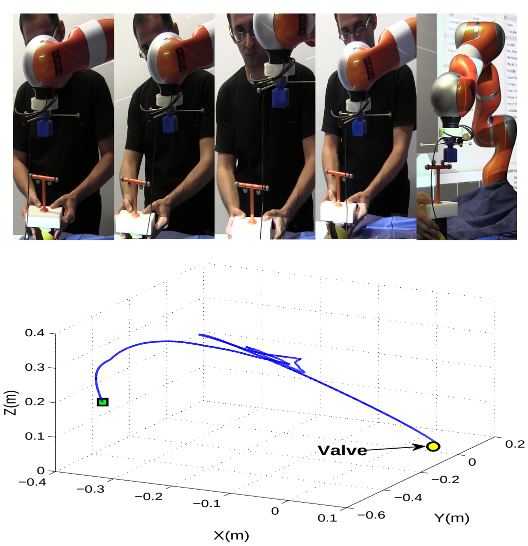

Fig. 8. Set of images from different steps of the experiment (top), the recorded positions of the end-effector supervised by tuned RFDM (bottom). The effect of RFDM can be seen in the middle of the trajectory while the valve is oscillating.

VIII. CONCLUSIONS

We proposed a hierarchical learning approach to deal with the challenging task of autonomous valve turning. We used kinesthetic teaching based on imitation learning to create trajectories from demonstrations. Then we developed a RFDM system to improve the autonomy in the task persistently. This RFDM system evaluates the dynamic behavior of the system and regulates the robot's movements reactively. Furthermore, we used the expert knowledge and optimization algorithms to tune the proposed RFDM based on apprenticeship learning. We successfully applied the proposed approach to the task of valve turning.

REFERENCES

[1] Persistent Autonomy through learNing, aDaptation, Observation, and Re-plAnning (PANDORA). http://persistentautonomy.com/
[2] D. Ribas, N. Palomeras, P. Ridao, M. Carreras, and A. Mallios, Girona 500 AUV, from Survey to Intervention, IEEE/ASME Transactions on Mechatronics, Vol. 17, 2012, 46-57.
[3] A. Albu-Schffer, C. Ott, and G. Hirzinger, A Unified Passivity-Based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots. Int. Journal of Robotics Research, Vol. 26, 2007, No 1, 23-39.
[4] G. Schreiber, A. Stemmer, and R. Bischoff, The Fast Research Interface for the KUKA Lightweight Robot, IEEE ICRA, 2010.
[5] M. A. Abidi, R. O. Eason, and R. C. Gonzalez, Autonomous Robotic Inspection and Manipulation Using Multisensor Feedback, IEEE Computer Journal, 24(4), pp. 17-31, April 1991.
[6] D. A. Anisi, E. Persson, and C. Heyer, Real-world demonstration of sensor-based robotic automation in oil & gas facilities, Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ Int. Conf. on, pp. 235–240, 2011.
[7] J. H. Ryu, Control of Underwater Manipulators Mounted on a ROV Using Base Force Information, IEEE Int. Conf. on Robotics and Automation (ICRA), Vol. 4, 2001, pp. 3238-3243.
[8] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, Robot programming by demonstration, in Handbook of Robotics, B. Siciliano and O. Khatib, Eds. Secaucus, NJ, USA: Springer, 2008, pp. 1371-1394.
[9] A. J. Ijspeert, J. Nakanishi, and S. Schaal, Trajectory formation for imitation with nonlinear dynamical systems, inProc. IEEE Intl Conf. on Intelligent Robots and Systems (IROS), Maui, USA, 2001, pp. 752-757.
[10] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction, ser. Adaptive computation and machine learning. Cambridge, MA, USA: MIT Press, 1998.
[11] A. Carrera, S. R. Ahmadzadeh, A. Ajoudani, P. Kormushev, M. Carreras and D. G. Caldwell, Towards Autonomous Robotic Valve Turning, Journal of Cybernetics and Information Technologies (CIT), vol. 12, no. 3, pp.17-26, 2012.
[12] S. Calinon, F. Guenter, and A. Billard, On learning, representing and generalizing a task in a humanoid robot, IEEE Trans. on Systems, Man and Cybernetics, Part B, vol. 37, no. 2, pp. 286298, 2007.
[13] P. Kormushev, D. N. Nenchev, S. Calinon, and D. G. Caldwell, Upper-body Kinesthetic Teaching of a Free-standing Humanoid Robot, IEEE Int. Conf., on Robotics and Automation (ICRA), 2011, pp. 3970–3975.
[14] L. X. Wang, A Course in Fuzzy System and Control, 2nd Edition. Prentice-Hall International, Inc., 1997, pp. 89-150.
[15] R.Y. Rubinstein and D.P. Kroese, The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning, Springer Verlag, 2004.
[16] P.T. De Boer, D.P. Kroese, S. Mannor, and R.Y. Rubinstein, A tutorial on the cross-entropy method, Annals of operations research, Vol. 134, no. 1, pp. 19-67, Springer, 2005.
[17] N. Hansen, The CMA evolution strategy: a comparing review, Towards a new evolutionary computation J., pp. 75-102, 2006, Springer.
[18] P. Brachetti, M. De Felice Ciccoli, G. Di Pillo, S. Lucidi, A new version of the Price's algorithm for global optimization, Journal of Global Optimization, vol. 10, no. 2, pp. 165-184, 1997, Springer.
[19] Video accompanying this paper, available online, http://kormushev.com/goto/ICRA-2013